What is claimed is:

1. A method of searching a query through distributed directory servers organized as a forest having a plurality of entries, each entry represented by a unique identifier, comprising the steps of:

generating a topology cache of the distributed directory servers;

storing the topology cache at each server; and

upon receiving, at one server, a request from a directory client associated with that server, sending a copy of the topology cache to the client to determine the identity of each directory server that can return answers to the query.

2. The method as defined in claim 1 wherein the topology cache includes subordinate and superior knowledge references associated with each directory server, defining its neighboring directory servers.

3. The method as defined in claim 1 wherein the method further comprises the step of:

forming, at the client, a distributed query evaluation plan *P* for a given query *Q*, based on the topology cache *T* sent to the client, where

$$P = Q_{S1}@S1 \cup \ldots \cup Q_{Sk}@Sk,$$

where *S1, ..., Sk* are defined as relevant directory servers for query *Q* extracted from the topology cache *T,* and Q_{s1} is the particular query for use with server *S1*

4. The method as defined in claim 3 wherein the method comprises the further step of:

generating a schedule for contacting the relevant servers included in the distributed query evaluation plan.

5. The method as defined in claim 1 wherein the query associated with the directory client comprises an LDAP query.

6. The method as defined in claim 3 wherein the query comprises an LDAP query and the distributed query evaluation plan comprises the steps of:

sending the original query to the directory server that manages the query's base-entry-DN; and

sending modified queries to all other relevant servers identified in the topology cache, the original query modified by replacing either the base-entry-DN, the scope, or both the base-entry-DN and the scope.

7. The method as defined in claim 4 wherein the query comprises an LDAP query and the scheduling step comprises contacting all relevant directory servers in parallel.

8. The method as defined in claim 4 wherein the query comprises an LDAP query and the scheduling step comprises contacting subsets of the relevant directory servers in parallel.

9. The method as defined in claim 4 wherein the query comprises an LDAP query and the scheduling step comprises contacting each relevant directory server in sequence.

10. The method as defined in claim 1 wherein the query comprises a complex query.

11. The method as defined in claim 10 wherein the complex query comprises a hierarchical query including either multiple-base-DNs, existential queries, conditional queries, or any combination thereof.

12. The method as defined in claim 10 wherein the complex query comprises an aggregate query including at least numeric macro conditionals.

13. The method as defined in claim 3 wherein the step of forming a distributed query evaluation comprises the step of generating a distributed query plan tree *PT*, comprising a root node, defined as the distributed evaluation query plan *P*; and

a plurality of non-root nodes comprising a first set of server query nodes *QSi* for relevant server *Si* in the topology cache *T*, and a second set of conditional nodes, the conditional nodes including existential queries and macro ("if") conditionals, wherein server queries are defined as AND nodes and conditional nodes are defined as OR nodes.

14. The method as defined in claim 13 wherein in evaluating the query plan tree, the following steps are used:

a) evaluating all existential queries at the conditional nodes;

b) expanding all macro conditionals to general query expressions; and

c) evaluating the server query nodes.

15. The method as defined in claim 4 wherein the step of generating a schedule for contacting the relevant servers includes

generating a distributed query plan tree *PT*, comprising

a root node, defined as the distributed evaluation query plan *P*; and

a plurality of non-root nodes comprising a first set of server query nodes *QSi* for relevant server *Si* in the topology cache *T*, and a second set of conditional nodes, the conditional nodes including existential queries and macro ("if") conditionals, wherein server queries are defined as AND nodes and conditional nodes are defined as OR node;

evaluating said distributed query plan tree to determine the different types of nodes; and

scheduling evaluation of conditional nodes before server query nodes.

16. The method as defined in claim 15 wherein in evaluating conditional nodes, the method further comprises the steps of

evaluating existential nodes prior to evaluating macro conditional nodes.

17. The method as defined in claim 16 wherein in evaluating existential nodes, the method comprises the steps of

creating a cache of all existential nodes in a query;

determining the answer to each existential query; and

populating the existential query cache with the determined answers.